NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

USING MULTIMEDIA METADATA TO IMPROVE NETWORK EFFICIENCY

By

Joe Blocker

September 2012

Thesis Advisor: Mathias Kolsch
Second Reader: Murali Tummala

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2012	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Using Multimedia Metadata to Improve Network Efficiency			**5. FUNDING NUMBERS**
6. AUTHOR(S) Joe Blocker			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ______N/A______.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

A growing amount of multimedia information exists online, commonly referred to as the multimedia explosion. Many research efforts are focused on the concept that computing is ubiquitous and that accessing network resources is not the challenge, but rather finding the right object is. However, America's military and other first responder organizations often find themselves in austere environments where access to network resources is scarce, and where every bit transmitted has to count. In this thesis we design a unique *Lead-Me* protocol that addresses both network access and finding the right data; but focuses on maximizing network efficiency by utilizing metadata information commonly found within multimedia files. We start by exploring other techniques commonly used to network efficiency, and then move to develop a protocol that fills the gaps. We use an intelligent middleware server that the client communicates to, direction-of-travel-aligned bounding boxes and mashup technology to reduce the size of the file the client receives as a response, and optimization techniques to prevent the client from receiving redundant files. We show an increase in efficiency of over 99% by using the middleware server, and an increase of 11% using the optimization techniques.

14. SUBJECT TERMS Multimedia, Networking, Bandwidth, Geospatial, Efficiency, Austere Environment			**15. NUMBER OF PAGES** 50
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

USING MULTIMEDIA METADATA TO IMPROVE NETWORK EFFICIENCY

Joe D. Blocker
Major, United States Marine Corps
B.S. University of Maryland 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2012

Author: Joe Blocker

Approved by: Mathias Kolsch
Thesis Advisor

Murali Tummala
Second Reader

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A growing amount of multimedia information exists online, commonly referred to as the multimedia explosion. Many research efforts are focused on the concept that computing is ubiquitous and that accessing network resources is not the challenge, but rather finding the right object is. However, America's military and other first responder organizations often find themselves in austere environments where access to network resources is scarce, and where every bit transmitted has to count. In this thesis we design a unique Lead-Me protocol that addresses both network access and finding the right data; but focuses on maximizing network efficiency by utilizing metadata information commonly found within multimedia files. We start by exploring other techniques commonly used to network efficiency, and then move to develop a protocol that fills the gaps. We use an intelligent middleware server that the client communicates to, direction-of-travel-aligned bounding boxes and mashup technology to reduce the size of the file the client receives as a response, and optimization techniques to prevent the client from receiving redundant files. We show an increase in efficiency of over 99% by using the middleware server, and an increase of 11% using the optimization techniques.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First I would like to thank my family for their support and dedication, not only to this thesis, but to 20 years of the Marine Corps. Without their constant willingness to sacrifice comfort for necessity, this endeavor would not have been possible.

Next, I would to thank the Marine Corps for providing me a lifetime of opportunity and self-improvement. I would not have been able to make the journey from a young, uneducated Private First Class to a Major with a Master's Degree without the support of Corps.

Professor Mathias Kolsch, thank you for your guidance and flexibility in this thesis. The process was both challenging and rewarding, and there is more I learned from you than any single document can reflect. The work in conjunction with the STEMA project, although ultimately not referenced here, was critical in developing the concepts that made this document possible. Jon in the vision lab also contributed many suggestions that helped push me along towards success.

Lieutenant Kima Banks, your work with databases often directly supported my research. In the process, we helped each other along the way and became friends in doing so.

Major Ben Hernandez, my study partner. Working with you taught me more than any other study sessions, thank you for always being there for me. Your friendship was invaluable.

Dr. Peter Denning, Professor Craig Martell, Commander Al Schaffer, Loren Peitso, Professor Tom Otani, Professor Hal Frederickson, Professor Ponte Stanica, and several other instructors were critical in my success. Thank you all for your work and service.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In this chapter, we briefly discuss the requirement for network efficiency, and the three factors that affect network efficiency when dealing with multimedia: the high availability and demand of multimedia, the current state of network resources, and the increasing demand for higher visual quality of multimedia files required to support larger display sizes.

A. REQUIREMENT

Military and first responder organizations are often called to respond to areas where bandwidth is limited—austere environments such as forward military operating areas or disaster areas where local infrastructure is disabled or otherwise not available. During military operations in Iraq and Afghanistan, the only data communication links were what the military brought with them, and likewise during the disaster recovery operations for the tsunami recovery in Thailand in 2004 and hurricane Katrina operations in 2005, the only resources available were what the first responders brought with them. These systems are frequently low-bandwidth systems such the Enhanced Position Location Reporting System (EPLRS) which the military uses for data communication to the lower tactical units which has a current capability of 486 kb/s(www.marcorsyscom.usc.mil, 2012).

B. AVAILABILITY

In recent years, there has been a multimedia explosion, with images, movie clips, and audio files from a plethora of sources being publicly available via multitudes of personal, commercial, and military means. Satellite imagery repositories are growing and updating consistently, to include Google's and Microsoft's commercial web-based systems. Unmanned Aerial Reconnaissance can provide up-to-the-minute imagery of battlefield situations, and even public sources such as Facebook, Photobucket, and other social networking sites have media repositories that would overwhelm network resources if they are not moderated.

> Improvements in network bandwidth along with dramatic drops in digital storage and processing costs have resulted in the explosive growth of multimedia (combinations of text, image, audio, and video) resources on the Internet and in digital repositories. (Christel, 1999)

Since 1999, the demand for multimedia and their production has accelerated even further with the growth of digital photography, smart phones, and aerial surveillance. In the midst of this high availability of multimedia content, the problem has become one of searching, sorting, and filtering through the media to find the right medium. This problem is described as "unmanageable without fine grained computerized support" (Garcia et al., 2008). This support is generally found in the form of metadata – information about the media – that is then parsed into databases for ease of searching, sorting, and filtering. The problems of managing media are increased when operating from a low-bandwidth austere environment.

C. NETWORK RESOURCES

Network resources have also grown explosively, with high speed internet at the home or workspace common, wireless high speed networks nearly ubiquitous, and even mobile data network speed is capable of high speed networking "Ubiquitous or pervasive computing defines a new paradigm for the twenty first century."(Ye et al., 2009) Additionally, the use of satellite technology allows (albeit slower) network resources to be deployed to austere environments for both personal and professional use. The majority of modern network protocols and technology are developed for these high-speed users, leaving users in austere environments with mismatched bandwidth and technology.

The challenge for low-bandwidth users is to find the right media that can be received over the low-bandwidth channel in a timely manner. The results from a database search may include several hundred files, and the result set alone could overwhelm a low-bandwidth system.

D. DISPLAYS

The displays today are also advancing, with high definition resolution becoming the accepted standard. Home theater projection is available in most technology stores, and mobile devices are getting larger to support higher display rates.

Users in austere environments, however, only have the resources they carry with them, and therefore the display sizes will likely be limited to laptop or handheld devices. Full HD quality may not be required or supported, and therefore reducing the pixel quality of the media

E. THESIS OVERVIEW

The purpose of this thesis is to find ways to support users who are in austere environments, such as military members on deployment, relief agency workers in disaster recovery environments, and even recreational users in the wilderness on limited bandwidth networks. We demonstrate techniques and protocols that use multimedia metadata to improve network efficiency in these austere environments. We begin by discussing other work in this area in Chapter II. Chapters III and IV cover what we want to achieve and how we go about doing it. Chapter V discusses the results of the experiment.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PREVIOUS AND RELATED WORK

In this chapter, we review several of the most common techniques for increasing network efficiency. There are many techniques; many focus on the idea that transmitting fewer bits per data allows more data to be transmitted over a fixed bandwidth channel. Others use metadata to identify the media before download to increase efficiency. Finally, we will look at several ontological techniques to improve the use of metadata in the task of increasing network efficiency.

A. TRANSMISSION

1. Compression

One of the techniques for reducing the number of bits transmitted per data is to use compression—representing data with less data; this is also a popular technique in coding theory and in fact was introduced early in the computer science lifespan as the Huffman Code. More modern advances include several types of compression, including naïve compression, that is completely unaware of what the bits represent and simply finds ways to reduce the number of bits, and content aware compression that knows what the bits represent and uses some artificial intelligence to eliminate bits. These techniques can also be subdivided into 'Lossy' techniques and 'Lossless' techniques – where lossless compression allows full recovery of the original data, bit for bit, and lossy compression throws away bits that can never be recreated. Typically, naïve compression techniques are lossless, while content aware techniques are lossy, basing their compression scheme on the type of media being compressed, and a programmed awareness of what human perception is capable of detecting, and then discarding the segments of the file that will not be missed by human perception.

Lossless compression techniques for multimedia, as seen in Figure 1, include Huffman, arithmetic decomposition, Lempel Ziv, and run length techniques (Furht, 1995).

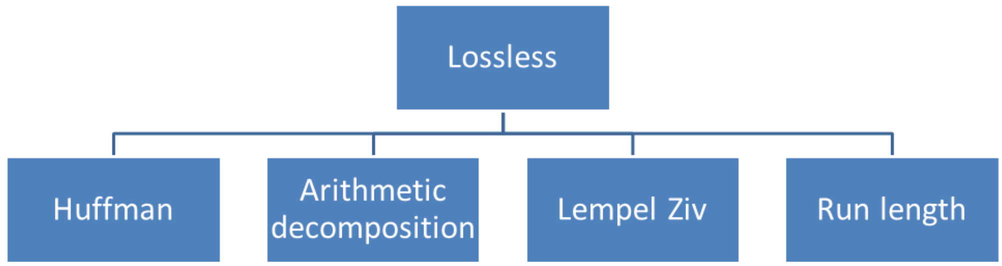

Figure 1. Lossless Compression Techniques for Multimedia (From Furht, 1995)

There are many lossy methods for compressing multimedia, which can be seen in Figure 2. This chapter will focus briefly on the hybrid techniques.

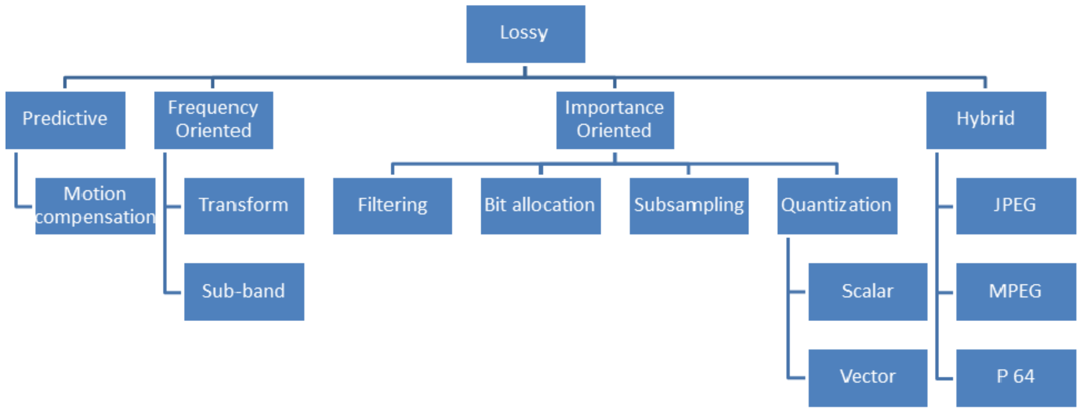

Figure 2. Lossy Compression Techniques for Multimedia (From Furht, 1995)

JPEG compression involves four modes – sequential, progressive, lossless, and hierarchical. The sequential and progressive modes both use discrete cosine transforms (DCT) techniques, while the lossless JPEG uses a predictive algorithm rather than DCT. Hierarchical JPEG creates a set of images at various resolutions.

The basic idea to video compression is to compute the difference between frames and to store or transmit only the differences – the idea being that most of the pixels in a

frame are constant between frames in a scene (Petrakis, ND). MPEG compression has several standards that use similar techniques. An initial frame (I-Frame) is encoded using JPEG compression. Within that frame, motion blocks are identified and their location within the next frame is computed in one of two ways: predictive coding or backwards coding (P-Frame or B-Frame). P-Frames transmit the motion block's predicted location from the I-Frame and transmit the vector for the next location of the motion block, while B-Frames use both a previous and future block (either I-Frame or P-Frame) to encode the motion block's location vector. The advantages of P-Frame encoding is faster decode times, because each frame is decoded in order, while the advantage of B-Frame encoding is higher compression rates. The best option for low bandwidth environments is a higher B-Frame rate encoding – even though more data is downloaded before playback begins, the overall data transmitted is lower.

2. Streaming

Another technique for improving network efficiency is streaming data, in which multimedia data is broadcast from a streaming server to a client for immediate playback. There are several protocols for streaming media, including the Real Time Streaming Protocol (RTSP) and other protocols for controlling the media (pause, play, search, etc.) such as the Realtime Control Protocol (RCP).

The basic problem with streaming data in low-bandwidth environments is that demand is increasing for higher display resolution and audio quality, which is increasing the bandwidth requirement. One technique for managing network efficiency is to use layering, where a base layer is transmitted with low resolution and audio quality, and as system demand and network bandwidth allow, additional layers are transmitted until either all layers are being transmitted, the client acknowledges that it is using its maximum resolution, or the network bandwidth is saturated. These systems "select the optimal scalability options for resource-constrained networks"(Lee et al., 2010). Specific implementations of layering technology include Scalable Video Coding (SVC) (Lee, 2010) and Dynamic Adaptive Streaming over HTTP (DASH) (Stockhammer, 2010).

3. Progressive Download

Progressive download is a technique similar to streaming where playback is initiated prior to the full availability or download of the media file. The primary difference between streaming and progressive download is that in a streaming media file, the content is not saved, making it difficult to steal raw content (Mow, 2007). The advantage that progressive download provides is rapid access to viewing the media; the disadvantage being that there is a possibility of inefficiency if the user elects to stop viewing before completion of the media – any data progressively downloaded but not viewed is wasted (Stockhammer, 2011).

4. Mashup

Mashup is a technique where several media file showing the same event are combined into a single file, optimally taking the best segment from each of the available files showing the event (Shrestha et al., 2010). This single file is then transmitted in lieu of the several original files. This improves network efficiency at the possible risk of losing intelligence as the mashup may not select a segment from a file that has what the client is looking for, based on the mashup servers selection criteria.

B. METADATA

The other elements of a multimedia file that are used to improve the efficiency of the network utilization include descriptive elements of a multimedia file, called metadata. Metadata are leveraged to reduce the data sent across the network.

> Ubiquitous multimodal sensors capture visual information in the form of moving and still pictures. In many applications, visual information needs to be organized according to relevance or compared with examples in a networked database to provide service to users. In either case, it is necessary to index the visual information based on its perceptual content and extract multimedia documents. (Ye et al., 2009)

1. Temporal

Temporal data is either relative or real world. Relative temporal metadata is used to select parts of a multimedia file to transmit. If an event is known to occur at a certain

point into a file, and is known to last for a certain length, then it is possible to select only the relative portion of the file that contains the period from the event start time to the event end time. The same can be true for still image files that are tagged with real world temporal metadata, especially in repetitive capture situations—the real world capture time can be used to narrow the number of images transmitted from an image store.

2. Content

> In recent years there has been a rapid increase in the size of video and image databases. Effective searching and retrieving of images from these databases is a significant current research area. In particular, there is a growing interest in query capabilities based on semantic image features such as objects, locations, and materials, known as content-based image retrieval. (Mallepudi et al., 2011)

The media itself also contains information that can be used to improve the network efficiency. The resolution and frame rate (also called temporal resolution (Lee et al., 2010)), or the compressed bit rate, can be used to select files that are appropriate for a certain display. Additionally, the media metadata is required for selecting the appropriate layers in the layering technique above.

Another system for describing the content of media is the multimedia content description standard developed by the Motion Pictures Expert Group (MPEG), called MPEG-7. MPEG 7 uses a standard format for all metadata entries, and is fully customizable to allow any content description field to be created based on the users needs.

The MPEG-7 standard has four specifications. First is the descriptor, which is the actual metadata about the media. Next is the description scheme, which consists of various elements that are either descriptors or other description schemes. MPEG-7 also defines a description definition language, which is XML based. Finally, MPEG-7 defines a scheme for coding its information for transmission (Martinez, 2004).

MPEG-7 starts by using descriptors, which are fields that contain the low level data about media. These elements are stored in various description schemes, along with relationships between these elements. Possible descriptors include the sensor location,

the real world start time, the location the media is showing, and the person or people being shown in the media. These descriptors are stored in different description schemes. There is a description scheme for sensor data that contains the sensor location descriptor, a temporal scheme that contains the start time descriptor, a geospatial scheme that contains the location descriptor, and a person scheme that contains the person descriptor(s). Finally there is a description scheme for the entire scene that contains these other description schemes and the relationship they all share. The result is a description scheme from which a user can ascertain that at a given time a given sensor recorded media of a certain person at a certain place. This data could then be sent to other users who would be able to use this information as they needed. This data transmission can be either the binary data as defined as a specification in MPEG-7, or the raw XML data used for human-readability. It is also significant that the MPEG-7metadata can be transmitted on its own, without any requirement for the associated multimedia to be transmitted with it. Four practical applications that use the MPEG-7 standard are briefly discussed in Section II.C, "Ontologies."

3. Media

Multimedia files also include in addition to the actual media – data about the media that is called metadata. Possible metadata fields include the sensor data, including location of the sensor, orientation, what types of bands are captured by the sensor, and the type of media captured by the sensor. Other metadata is descriptive in nature, referred to as content description metadata.

Using the content description metadata is also a way to improve the network efficiency. Location data is frequently stored within images, and tags often identify who is in an image or what is taking place in an image – person data and event data. A powerful combination of these can be used to filter images/media files from media stores for efficient use.

C. ONTOLOGIES

> The most quoted paper on the definition of ontology today is by Gruber (1992) where he said "An ontology is an explicit specification of a

> conceptualization." What is mostly over-looked in this definition is what was said in the rest of the paper. "The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems, what exists is exactly that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. A common ontology can serve as a knowledge-level specification of the ontological commitments of a set of participating agents. A common ontology defines the vocabulary with which queries and assertions are exchanged among agents."(Santini et al., 2010)

The challenge in efficiently using content description is the semantic quality of content description. This is referred to as "Bridging the semantic gap."(Bao et al., 2010) To this end, several ontologies have been created for various description schemes. Not only are there many schemes for categorical data (where, when), but proposed schemes exist for the semantic properties as well (what, how, and who). The ontology structure is also "exploited to encode semantic relations between concepts."(Bertini et al., 2010)

Several schemes exist for defining the format of these ontologies. The objective is to develop a scheme that is both machine understandable and human readable for parsing and searching. Several ontologies exist for many specific areas of interest. The Geographic Markup Language (GML) defines an ontological scheme for describing the geospatial features images. Other ontologies exist, based on the MPEG-7 content description standard, for various applications. See Table 1.

	Hunter	**DS-MIRF**	**Rhizomik**	**COMM**
Foundations	ABC	none	none	DOLCE
Complexity	OWL-Full	OWL-DL	OWL-DL	OWL-DL
Coverage	MDS+Visual	MDS+CS	All	MDS+Visual
Applications	Digital Libraries, e-Research	Digital Libraries, e-Learning	Digital Rights Management, e-Business	Multimedia Analysis and Annotation

Table 1. Summary of different MPEG-7 based Multimedia Ontologies (Trony, et al., 2007)

Without these and other specifications, the flexibility of the MPEG-7 standard is also its weakness, as described by Troncy, et al.:

> The flexibility of MPEG-7 is therefore based on allowing descriptions to be associated with arbitrary multimedia segments, at any level of granularity, using different levels of abstraction. The downside of the breadth targeted by MPEG-7 is its complexity and its ambiguity. (Troncy, et al., 2007)
>
> For example, very different syntactic variations may be used in multimedia descriptions with the same intended semantics, while remaining valid MPEG-7 descriptions. Given that the standard does not provide a formal semantics for these descriptions, this syntax variability causes serious interoperability issues for multimedia processing and exchange. (Troncy, et al., 2007)

III. THEORY AND METHODS

This thesis demonstrates that network efficiency can be improved by using an intelligent middleware server that processes a database result set based on information from the media metadata. The key focus is improving efficiency in communication to and from a client computer. To address this, we discuss the use and application of geospatial metadata that has been parsed into an appropriate geospatial database. In particular, we discuss simple database queries, database queries that have their result sets processed and filtered by an intelligent middleware server, and database queries that not only utilize the intelligent middleware server, but also employ optimization techniques for further improving network efficiency. We also discuss the importance that image density plays in improving network efficiency, the importance of a direction-of-travel oriented bounding box, as opposed to a north aligned bounding box, and we discuss a specific protocol for client/server communication.

A. BASE CASE

The base case for this thesis is a simple geospatial database query protocol that transmits a database query message to a database and receives the result set of that query in return. Figure 3 provides a visual representation of this query process.

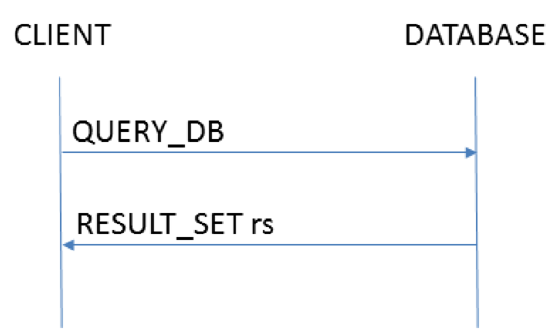

Figure 3. Database query process

The *image density* (d) is the proportion of images to the *area of coverage* (a), in this paper image density is expressed in images per square kilometer. The *result set size* (r) is dependent on both the *image density* and the *area of coverage:* e.g.,

r=d*a

So clearly, in areas of high *image density*, or queries with a large *area of coverage*, the *result set size* will potentially be very large. In Section IV.A, we specify a region with an area of 10062 square kilometers with 100,000 images in the area. Deriving from the formula above, this gives an image density of 9.94 images per square kilometer. Also in Section IV.A, we specify an area of coverage for bounding box of 12 square kilometers (3 km by 4 km rectangle). Using the derived image density of 9.94 images per square kilometer and an area of coverage of 12 square kilometers, we expect a result set size of nearly 120 (119.25).

C. METHODS

In this thesis, we design a protocol that uses an intelligent middleware server to process the result set of the database query improves network efficiency to and from the client. It assumes that network resources between the server and database are abundant and are not an issue. Figure 4 provides a visual representation of this process.

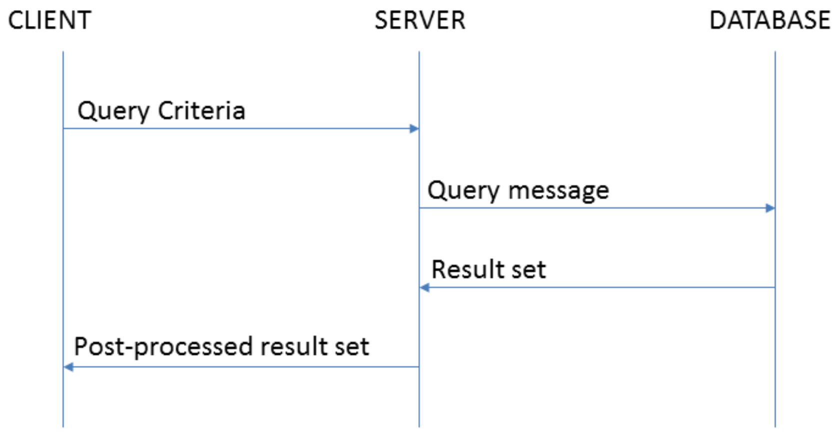

Figure 4. Database query process with intelligent middleware server

We took this basic design and modified it to utilize the Lead-Me protocol described in Section III.C.1, "Lead-Me Protocol." The middleware server application has three elements to address: query criteria, query message, and post-processed result set.

The query criteria are sent to the server via the Lead-Me protocol that was designed for this thesis. The first message contains static data. The server saves this data as client state to avoid transmitting duplicate data with every message. This requires a persistent connection to keep state, but because the application is designed to make continuous requests, the overhead of the persistent connection was deemed acceptable. The second message contains the geospatial data elements that are unique to each specific query message. See Section II.C.1, "Lead-Me Protocol" below for specific details.

The function of the middleware server includes the ability to process geospatial data into a Lead-Me bounding box, which is direction-of-travel aligned, and generate a properly formatted database query with that bounding box. The middleware server also includes an image processor that is available to create a mash-up of all images from the result set into a single image file. The image processor rotates and crops the image so it is oriented along the direction of travel and contains only the user requested area defined by the lead-me bounding box. Figure 5 provides a visual representation of the Lead-Me process.

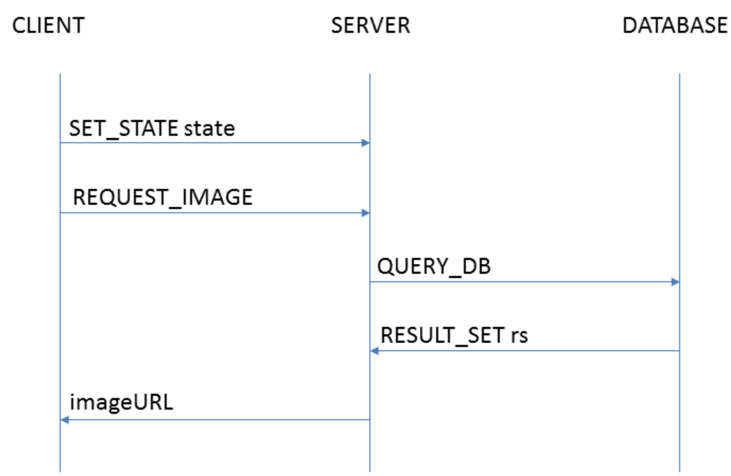

Figure 5. Lead-Me database query process with intelligent middleware server

Without the direction-of-travel aligned bounding box (e.g. using north aligned bounding boxes), the result set would potentially include images that did not actually touch the desired lead-me bounding box. Figure 6 compares a north-aligned bounding box with a direction-of-travel-aligned bounding box, and clearly shows that in order to capture the entire lead-me region, the north aligned bounding box will query regions that are outside of the lead-me region (the shaded areas). In Figure 6, the lead-me bounding box (the checkered area) is set at a standard 3:4 aspect ratio and is tilted 25 degrees east of north. The lead-me bounding box has an area of 12 square units, while the north-aligned bounding box has an area of just over 21 square units. In this case, the lead-me bounding box has a savings of nearly 44% over the north-aligned bounding box. When compared using all angular variations for the direction-of-travel-aligned 3:4 aspect ratio bounding box, the average savings was computed as 39.57%

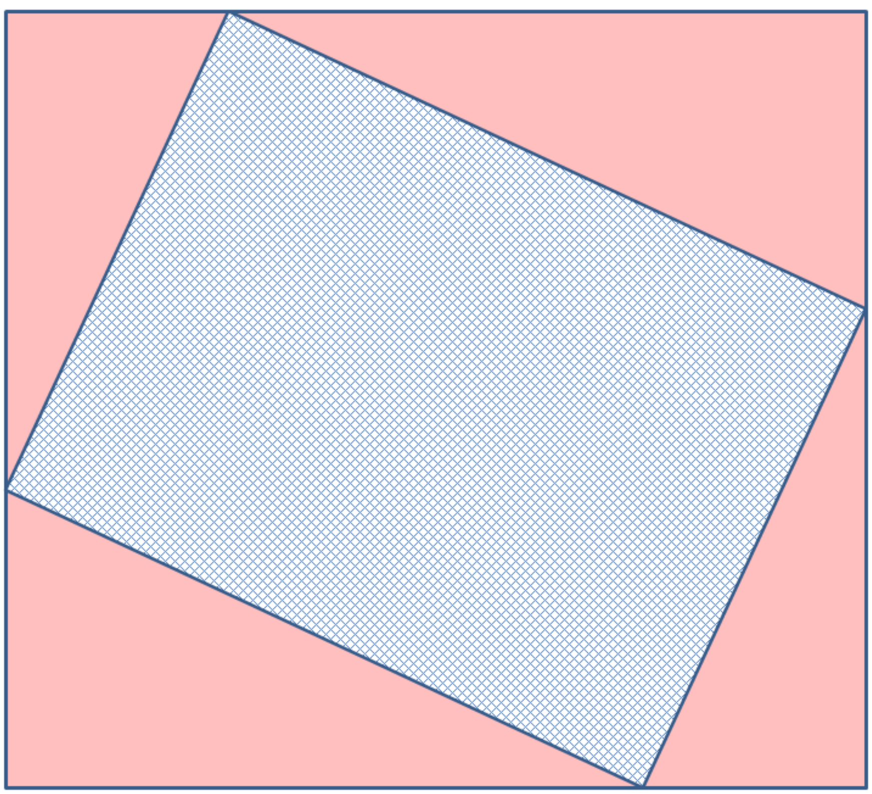

Figure 6. North-aligned bounding box vs. direction of travel aligned bounding box

The exterior red shaded area is the north-aligned bounding box, while the interior blue checkered area is the direction-of-travel aligned bounding box.

1. Lead-Me Protocol

The Lead-Me protocol is designed as a thin client protocol for communicating requests for imagery from a client to a server and returning a response to the client from the server. As a thin client protocol, the client does little processing, and simply sends data to the middleware server, which is where the actual query string is created and the query takes place. The protocol contains two messages: SET_STATE and REQUEST_IMAGE.

a. Set State

The SET_STATE message decreases network traffic by storing some constant values on the server rather than transmitting them with each REQUEST_IMAGE message. The message contains eight fields: message type, bandwidth, display width, display height, bounding box width, bounding box depth, lead-me type, and lead-me factor. This results in seven of these fields (message type is sent with every message) being sent only once rather than with every REQUEST_IMAGE message.

The message type is a single character; in the SET_STATE message the message type is always set to ‘s’. The next seven values are all numeric values. The values for each field represent the following measurements: Bandwidth, display width and display height in pixels, bounding box width and depth in ground distance, lead-me type in integer format representing the type of lead-me to use – time or distance – and lead-me factor in time or distance depending on the lead-me type. The protocol does not specify the method for determining the contents of the SET_STATE message – programmatic techniques can be used to determine bandwidth and screen resolution, or they can be user defined via a user interface. Likewise the units of measure for the bandwidth, bounding box, and lead-me factor can be any applicable bandwidth, time, or distance measure, as long as the client and server agree what the unit of measure is. See Table 2.

SET STATE	
Content	Comment
Message type	‘s’ for set state
bandwidth	
display resolution width	pixels
display resolution height	pixels
bounding box width	
bounding box depth	
lead-me type	0: time / 1: distance
lead-me factor	

Table 2. SET_STATE message format

b. Request Image

The REQUEST_IMAGE message contains the data that the server uses to compute the bounding box coordinates. The server computes the bounding box from the client’s current location, heading, and velocity, along with the lead-me type and factor in the SET_STATE message, to find the center of the Lead-Me bounding box. Then the server uses the bounding box width and height from the SET_STATE message to compute the corners of the Lead-Me bounding box. The requirement is for the server to compute a box with four coordinate pairs that outlines a box oriented along the direction of travel. Table 3 contains the REQUEST_IMAGE message format.

REQUEST_ IMAGE	
Content	Comment
Message type	"r" for request image
Current position latitude	
Current position longitude	
Current heading	
Current velocity	

Table 3. REQUEST_IMAGE message format

2. Communication Between Server and Database

The communication between the server and the database is outside the scope of this thesis, as there are many different types of databases available with multiple

variations of geospatial database queries. The constraint is that the database query should return a set of images or image file locations that the middleware server can use. However, to better illustrate database operations, we briefly discuss our implementation details for the experiment in Chapter IV.

3. Image Processor Functions

The image processor specifications are: The image processor must accept a set of images or image file locations, a set of bounding box coordinates, and a desired image resolution, and after completion of processing, return an image or image file location that contains the image that has been mashed-up, rotated to match the direction of travel, cropped to include only the area contained by the bounding box coordinates and resized to match client resolution. This includes the implied capability to interpolate differences between mismatched image resolutions in the result set to process the final mashup. Additional specifications could include the ability to give preference to temporal selections (e.g. most recent or matching current time of day) and spectral selection (natural color, infrared, combinations, etc.). The implementation of the image processor that meets these specifications is outside the scope of this thesis.

4. Request Image Response Messages

There are three possible responses that the server can send to a REQUEST_IMAGE message, depending on the result set returned from the database. If the database query returns an empty result set, the server will send a message of *empty*. If the current database query result set is identical to the previous database query result set, the server will send a message of *no_change*. If neither of the previous messages is selected, the location of the file returned from the image processor is sent to the client for processing or download. Table 4 summarizes the request image responses.

RESPONSES	
Case	Message type
Database Result Set is empty	"*empty*"
Database Result Set equals the previous Database Result Set	"*no change*"
Default(not empty, not equals to previous)	URL of processed image

Table 4. REQUEST_IMAGE Response Messages

IV. IMPLEMENTATION

In this chapter we discuss the setup and conduct of the experiment that measures the network efficiency of a simple database query against a query using the lead-me protocol. We begin by explaining the assumptions made, and then proceed to explain the details of setting up the client and server applications. We close the chapter by explaining what we measure during the experiment and why.

A. ASSUMPTIONS

A geospatial database is required for this experiment. Because NPS has a pre-built instance already available, a MySQL server is used as the database server. The database is populated with 100,000 data points representing images each 1km by 1km. All images are randomly located in a box approximately 100km by 100km. Specifically, the box is located with a starting corner at 35N 120W and extended 1 degree north and east, for actual dimensions of 111.18 km north to south 89.94 km east to west on the north edge of the box, and 91.07 km east to west on the south edge of the box. This provides an average image density of 9.94 images/square kilometer.

There is no image processor used in this experiment. The image processing is simulated by selecting a single simulated image from the database result set and returning the single image, just as the image processor is expected to return a single image. The resolution of the returned image is set by the SET_STATE message explained below.

All experiments use the same route specified as simulated GPS input, through the database image field from south to north, starting at 35.0N, 119.2W and traveling in a straight line in a north by north-west direction (345 degrees heading). The starting latitude is set to 35.0N so that the route starts on the southern edge of the image box. The starting longitude and the heading are both random selections with no influence on the outcome of the experiment. New coordinates are generated every 30 meters to simulate a new image every second at a velocity of 90 km/hr. The route ends when the vehicle reaches the end of the image box, as determined by the coordinates above.

All experiments first sent a fixed SET_STATE message with a bandwidth of 100, display resolution set to 1200x1600, the lead-me factor set to ‘distance,’ and with a distance of 5 km, and a lead-me view box of 4km wide by 3km deep. Table 5 shows the SET_STATE values used for the experiment.

SET STATE		
Content	Comment	Value
Message type	"s" for set state	"s"
bandwidth	MB/S	100
display resolution width	pixels	1600
display resolution height	pixels	1200
bounding box width	km	4
bounding box depth	km	3
lead-me type	0: time / 1: distance	1
lead-me factor	km or min	5

Table 5. SET_STATE values used

B. SETUP OF THE EXPERIMENT

The programs that run the experiment are written in Java. The experiment consists of two distinct programs that communicate via Java sockets (TCP).

1. Client Program

The client program is designed to accept a geospatial location from the simulated GPS and to send that data via the Lead-Me protocol messages to the server application. After the server completes its processing, the client program receives a message back from the server with a URL for a post-processed image file, or an *empty* or *no change* message if applicable.

2. Server Program

The server program was designed to accept Lead-Me protocol messages from the client application, and compute the coordinates of the Lead-Me bounding box. Then the server application queries the geospatial database for an intersection between the image coordinates and the Lead-Me bounding box coordinates. The server receives the result set from the database and simulates image processing by selecting a single image from

the database to return to the client program. Two variations of the server program were tested—the first, referred to as unoptimized, returned a single image file for every database query that had at least one image in the result set; the second, referred to as optimized, returned a *no_change* message if the result set of the current query was identical to the result set of the previous query.

3. Image Size and Resolution

Images from a Naval Postgraduate School unmanned aerial vehicle were sampled and the average file size for images from the onboard sensor was computed as 1228.83 kb. Each image from the sensor was captured at a resolution of 2452 x 2056 pixels. Comparing this resolution the client's desired image resolution of 1600 x 1200 pixels, the ratio of desired size to actual size was calculated as 38.09%. Using this ratio on the actual average file size yielded a desired average file size of 468.01 kb.

4. Number of Queries

For the traversal of the simulated GPS path through the image field, there were a total of 3705 distinct points queried to the database in each case.

C. MEASUREMENTS AND COMPARISON

There is one independent variable with three values for this experiment, representing the three levels of capability of the middleware server. The absence of the middleware server is the first value for this variable and is referred to as the base case. The presence of the middleware server in its unoptimized form is the second value and is referred to as unoptimized. The third value is the presence of the middleware server with optimization and is referred to as optimized.

The dependent variable that is measured and analyzed is the total amount of data received by the client. This is calculated as the average image file size times the number of image files received by the client. For the base case, this is the total number of image files returned by every query. For the unoptimized case, this is the number of queries for

which at least one image was returned. For the optimized case, this is the number of queries for which at least one image was returned, and for which the result set is not identical to the previous result set.

V. RESULTS AND DISCUSSION

A. RESULTS

In the base case, the total number of responses received by the client was 406,273 image files for a total transmission requirement of 185,681.5 Mb. This is the sum of every image in every data set resulting from every query, which reflects the simple database query method with no intelligent middleware server. In the unoptimized case, the total number of responses received by the client was 3590. The total transmission requirement for this solution is 1640.76 Mb. This reflects a single image being returned for every query for which there was at least one image available. The remaining 115 queries resulted in *empty* response messages being delivered to the client. In the optimized case, the total number of responses provided was 3194. This reflects a single image being returned for every query for which there was at least one image available, and for which the query result set was not identical to the previous query result set. With 3194 images returned, and 115 *empty* responses, there were 396 *no change* messages sent in the optimized case. The total transmission requirement for this solution is 1459.77 Mb.

The base case clearly required bandwidth that was orders of magnitude greater than any of the test cases, as shown in Figure 7. The ability to reduce the result set size from a large number (over 100) to a single returned image file location as seen in the base case resulted in savings of 99.088%. The savings will vary depending on the density of the images in a given region—greater image density will yield greater savings when using these techniques.

.

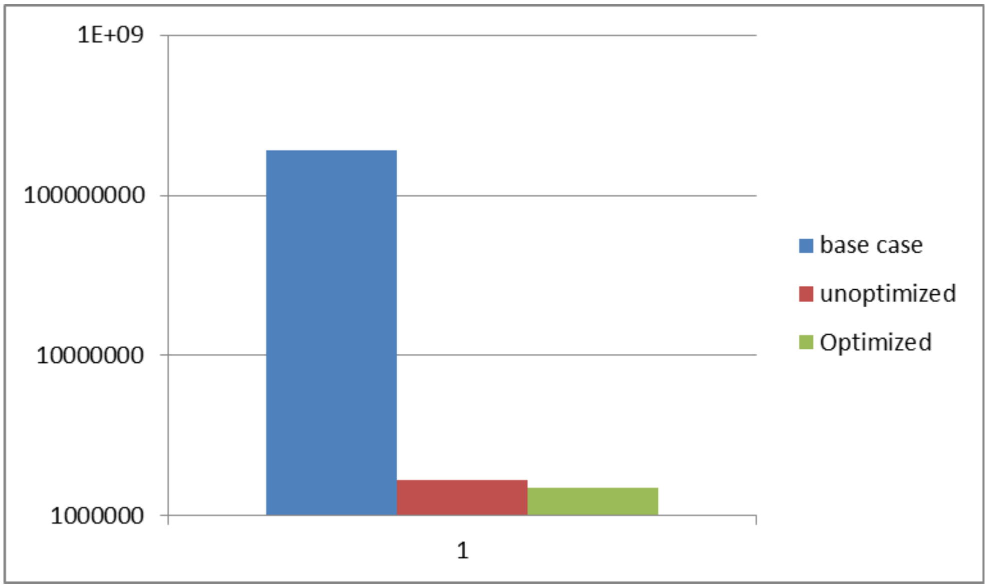

Figure 7. Base case requires orders of magnitude more bandwidth.

Additionally, the ability to eliminate transmission of duplicate images yielded a savings of 11.03%. This was a result of *no change* messages in the optimized case reducing the total transmission requirement from 1640.76MB to 1459.77MB. Also, the average result set size for each case was 113.16 images received by the client. The expected value for this independent variable, computed in Section III.A, was 119.25.

B. DISCUSSION

The stated task in this thesis was to achieve a result that demonstrated efficiency. The implied task in this thesis so far has been to achieve a result that is both sound (i.e., it contained elements that were not actually in the desired solution) and complete (i.e., it contained every possible correct element). Given the simple database query in the base case, we achieved a result that was sound and complete, but it was inefficient. By applying the techniques in the Lead-Me protocol, we achieved a result that is sound and complete, and at the same time, much more efficient that the simple database query.

All of these results are an improvement over various other current protocols that rely on north-aligned bounding boxes. As demonstrated in Section III.C, the direction-of-travel-aligned bounding box provides a savings of nearly 40% given the size of the

lead-me bounding box used in this thesis, and up to 50% if the lead-me bounding box were square rather rectangular. This savings is a result of the north-aligned bounding box providing results that are complete, but not sound. The areas in the north-aligned bounding box (the shaded area in Figure 6) represent areas outside of the direction-of-travel-aligned bounding box (the checkered area in Figure 6). This clearly represents areas where the provided solution does not fall into the desired solution, which is defined as unsound.

The resulting improvement from the base case to the unoptimized case is theoretical at this stage; while there is clearly a large degree of improvement shown, the result was based on a hypothetical image processor that uses already researched mashup techniques. The implementation of this image processor, as found in Section III.C.3, "Image Processor," is left as future work.

The optimization techniques found in the optimized case also resulted in a sound and complete solution. The simple elegance of that solution was to compare result sets from the database and eliminate any result set did not have at least one item different than the previous result set. Given that the previous result set was both sound and complete, there was no need to repeat it.

Additional improvements could be made to the optimized method by comparing the current and previous result sets in more detail. In that way, the size of the image received by the client machine would be reduced, as the image processor would only construct a new image from new elements in the current result set. The previous image would be transformed by a movement vector, and any pixels in the image no longer in the lead-me bounding box would be discarded. Then the new image would be transmitted to reestablish the soundness and completeness of the image presented to the user. However, this would require additional protocol elements, and would also require additional processing on the client end, which detracts from the "thin client" model described herein. Additional research is warranted to see if the gains in network efficiency warrant the increase in client side computing.

Additionally, comparing the geospatial coordinates of the current and previous lead-me bounding boxes would also increase optimization. In this technique, the database query formed by the middleware server could be designed to query only the difference between the bounding boxes. Much like comparing result sets in the previous paragraph, this technique would maintain soundness and completeness while reducing the amount of data sent to the client, but simultaneously increasing the processing required at the client. The potential extra benefit of this technique would be to reduce the size of the result set, potentially decreasing database, and image processing time.

VI. CONCLUSION

The purpose of this thesis is to find ways to support users who are in austere environments. We have shown that using an intelligent middleware server to process the result set based on multimedia metadata, specifically geospatial metadata in imagery, reduces the amount of bandwidth required to receive back a complete result set from a database query. We have also shown that using a few simple optimization techniques can even further reduce the amount of bandwidth required to send

A simple database query can yield a very large result set, and receiving this result on a low-bandwidth system in an austere environment can overwhelm the network resources. By inserting a middleware server between the client and the database that processes the result set using the multimedia metadata, we have reduced the bandwidth requirement from O(n) to O(1), where n is the size of the result set.

We have also shown that monitoring concurrent result sets for equality can further reduce the bandwidth requirement. In cases where two or more consecutive result sets are equal, the subsequent image transmissions are eliminated. In the results we find that this technique reduces the total bandwidth requirement by an additional 11 percent.

We have further shown that the use of the direction-of-travel aligned bounding box can reduce the size of the result set by nearly 40 percent. More significantly, the direction-of-travel aligned bounding box ensures that the result set is sound by eliminating rows that would be returned by a north-aligned bounding box but that are outside of the desired box.

Users in austere environments on low-bandwidth networks should utilize middleware servers, such as the Lead-me application server, to process their data using available metadata in order to reduce the bandwidth requirement for receiving sound and complete database result sets.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Bao, L., Cao, J., Zhang, Y., Li, J., Chen, M., & Hauptmann, A. (2010). Explicit and Implicit Concept-based Video Retrieval with Bipartite Graph Propagation Model. ACM Multimedia 2010, 939–942.

Bäuml, M., Fischer, M., Bernardin, K., Ekenel, H., & Stiefelhagen, R. (2010). Interactive Person-Retrieval in TV Series and Distributed Surveillance Video. ACM Multimedia 2010,1637–1638.

Bertini, M., D'Amico, G., Ferracani, A., Meoni, M., & Serra, G.(2010). Sirio, Orione and Pan: an Integrated Web System for Ontology-based Video Search and Annotation.

Bertini, M., D'Amico, G., Ferracani, A., Meoni, M., & Serra, G.(2010). Web-based Semantic Browsing of Video Collections using Multimedia Ontologies. ACM Multimedia 2010, 1629–1632.

Beecks, C., Driessen, P., & Seidl, T. (2010). Index Support for Content-Based Multimedia Exploration. ACM Multimedia 2010, 999–1002.

Changuel, N., Mastronarde, N., Schaar, M., Sayadi, B., & Kieffer, M. (2010). End-to-end Stochastic Scheduling of Scalable Video Over Time-Varying Channels. ACM Multimedia 2010, 731–734.

Christel, Michael G. (1999). Automated Metadata in Multimedia Information Systems: Creation, Refinement, Use in Surrogates, and Evaluation. Chapel Hill Morgan and Claypool.

Dasiopoulou, S. Tzouvaras, V. Kompatsiaris, I. & Strintzis, M. (2010). Enquiring MPEG-7 Based Multimedia Ontologies. *Multimedia Tools and Applications*, 46(331–370).

Furht, B (1995). A Survey of Multimedia Compression Techniques and Standards. Part I: JPEG Standard. *Real Time Imaging, 1*, 49–67 Retrieved from http://www.cse.unr.edu/~bebis/CS474/Handouts/SurveyCompression.pdf.

Garcıa, R, Tsinaraki, C., `Celma, O., & Christodoulakis, S. (2008). Semantic Multimedia Ontologies: Theory and Applications: *Multimedia Content Description using Semantic Web Languages*, Springer.

Hunter, J. (2001). Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. *International Semantic Web Working Symposium (SWWS)*, Stanford.

Jain, R., & Sinha, P. (2010). Content Without Context is Meaningless. ACM Multimedia 2010, 1259–1268.

Lee, J., Simone, F., Ramzan, N., Zhao, Z., Kurutepe, E., Sikora, T., Ostermann, J., Izquierdo, E., & Ebrahimi, T.(2010). Subjective Evaluation of Scalable Video Coding for Content Distribution. ACM Multimedia 2010, 65–72.

Mallepudi, S., Abhishikth, C., & Knapp, G. (2011). Material classification and automatic content enrichment of images using supervised learning and knowledge bases Multimedia on Mobile Devices 2011; and Multimedia Content Access: Algorithms and Systems V. Edited by Akopian, David; Creutzburg, Reiner; Snoek, Cees G. M.; Sebe, Nicu; Kennedy, Lyndon. Proceedings of the SPIE, Volume 7881, pp. 788113-788113-11.

Martinez, J., (2004). MPEG 7 Overview, retrieved from http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm#2.1_MPEG-7_Systems.

Mow, D., (2007). Streaming vs Progressive Download, Streaming Media, June/July, retrieved from http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Streaming-vs.-Progressive-Download-65076.aspx.

Petrakis, E., (ND). MultiMedia Compression, retrieved from http://www.intelligence.tuc.gr/~petrakis/courses/computervision/compression.pdf.

Rahman, A Hossain, A Kiringa, I & Saddik, A (ND). Towards an Ontology for MPEG-7 Semantic Descriptions retrieved from http://www.site.uottawa.ca/~kiringa/Papers/i2lor06.pdf.

Sánchez, Y., Schierl, T., Hellge, C., Wiegand, T., Hong, D., Vleeschauwer, D., Leekwijck, W. & Louédec, Y. (2010). iDASH: Improved Dynamic Adaptive Streaming over HTTP using Scalable Video Coding. Multimedia Systems Conference 2011, 257–264.

Santini, S., & Gupta, A. (2010). “Disputatio” on the use of Ontologies in Multimedia. ACM Multimedia 2010, 1723–1727.

Shrestha, P., With, P., Weda, H., Barbieri, M., & Aarts, E. (2010). Automatic Mashup Generation from Multiple-camera Concert Recordings. ACM Multimedia 2010, 541–550.

Stockhammer, T.(2011). Dynamic Adaptive Streaming over HTTP –Standards and Design Principles. Multimedia Systems Conference 2011, 133–143.

Troncy R., Celma O., Little S., Garcia R., & Tsinaraki C.(2007). MPEG-7 based Multimedia Ontologies: Interoperability Support or Interoperability Issue *1st Workshop on Multimedia Annotation and Retrieval enabled by Shared Ontologies*.

Ye, L., Ma, J. & Huang, R. (2009). Indexing and retrieval of ubiquitous visual information. Ubiquitous Multimedia Computing, (231-258). USA: Chapman and Hall/CRC.

http://www.marcorsyscom.usmc.mil/sites/cins/TCS/DATA%20RADIOS/EPLRS.html 7 September 2012.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California